

From ASI's continuing program of new product development...

# ASI

**ADVANCE SERIES**

**PROGRAMMING**

**SYSTEMS**

**DESCRIPTION**

*Advanced Scientific Instruments / 8001 Bloomington Freeway / Minneapolis, Minnesota 55420*

# **Advanced Scientific Instruments introduces a new programming package for the *ADVANCE* Series**

- **One-Pass EXTENDED FORTRAN II**
- **One-Pass Symbolic Macro Assembler—ASIST**
- **Automatic Operating System—MONITOR**
- **Mathematical Subroutine Library**
- **Utility Program Library**

These proven programming aids provide a complete package for use with the *ADVANCE* Series digital computing systems from ASI. These programming aids permit full utilization of all the *ADVANCE* Series systems, as these systems are fully upward compatible from the low cost 6020 to the larger, higher performance members of this Series.

In addition, ASI also offers complete programming support in the form of both programming assistance before and after delivery and special training courses in all phases of ASI program package. In addition, regularly scheduled classes and workshops are held at the ASI facilities in *ADVANCE* Systems.

The ASI's User's Group is organized to provide a convenient method of program interchange between all *ADVANCE* Series users.



## ADVANCE Series Extended Fortran II

ASI's ADVANCE Series EXTENDED FORTRAN II Compiler includes many extra features, in addition to the standard features of FORTRAN II. Included are many FORTRAN IV features and extended debugging capabilities. These features make the ADVANCE Series FORTRAN II a very powerful language for the solving of a wide variety of problems. The use of FORTRAN provides for rapid and easy programming and debugging of problems without learning the intricacies of the machine language.

Some of the more important features of the ADVANCE Series FORTRAN II Compiler are:

### SOURCE LANGUAGE FEATURES

- Complete Set of Standard FORTRAN II Statements
- In-line Assembly Language Coding
- FORTRAN IV Type Declaration Statements
- FORTRAN IV Logical Expressions and Statements
- FORTRAN IV Input/Output Statements
- FORTRAN IV DATA Statement
- Mixed Mode Arithmetic
- Complete FORMAT statement capabilities
- Typewriter and Paper Tape Input/Output Statements
- Use of special Field Terminator on formatted input
- Variable dimensions in subprograms
- Use of Hollerith Constants
- Use of up to nine Continuation cards

### DEBUGGING AIDS

- TRACE Statement
- Optional Memory Map
- PDUMP and DUMP Subroutines
- Comprehensive Error Diagnostics

### REAL TIME CAPABILITIES

- In-line Symbolic Assembly Language Coding
- Recursive Subroutine Calls

### OPERATIONAL FEATURES

- Dynamic Storage Allocation
- Program Linking
- One-pass Compilation from Source Language to Binary
- Optional Assembly Language Output
- Optional Listing of Source Language at Compile Time
- Flexible Use of Input/Output at Compile Time
- Operational in 4096 words of memory
- No magnetic tape units are necessary for compilation
- Subprograms may be compiled separately from main program
- Assembly Language subprograms may be called in FORTRAN

### Standard Fortran II Features

All of the features of standard FORTRAN II are included in the ADVANCE Series EXTENDED FORTRAN II Compiler. (Exception: the B, D, and I column 1 modal punches signifying Boolean, double-precision, and complex operations, are not allowed.) Source language programs written for other machines which do not use special features of the particular machine used, can be compiled and executed with ADVANCE Series FORTRAN II. Programs written for the ASI Compiler, that do not take advantage of some of the additional features listed, can be compiled and executed on other machines having standard FORTRAN II compilers.

### In-Line Symbolic Coding

Symbolic assembly language coding may be intermixed freely with FORTRAN II source statements. A subset of the ASI ADVANCE Series assembly language may be used. This allows the programmer to perform machine language functions not available in the FORTRAN II source language. It allows special functions, like those encountered in real time processing, to be coded as part of the FORTRAN II source program. No special control cards are needed to define the symbolic coding. Symbolic instructions may be placed freely in the FORTRAN II source program.

### Fortran IV Logical Capabilities

FORTRAN IV logical capabilities are provided. These include: type-declaration statements, logical operators, logical expressions, relational operators, logical assignment statements, and the logical IF statement.

### Mixed Mode Expressions

Mixed mode expressions are permitted in the ADVANCE Series FORTRAN II Compiler. In a mixed mode expression, both fixed point and floating point quantities are accepted.

### Format Specifications

ADVANCE Series FORTRAN II provides complete formatting capabilities. All standard FORTRAN II field specifications are handled by the ASI format scan routines. In addition, a special field terminator can be used in input data to terminate the input field and eliminate the necessity of counting columns when paper tape is used for input.

### Dynamic Storage Allocation and Recursive Subroutine Calls

Dynamic storage allocation is provided in a separate version



of the compiler. Each time a subroutine is called, using this feature, the unique storage area for that subroutine is assigned to the first storage available. Thus, all subroutines called on the same level will share the same storage area. This results in a significant storage saving in most cases. In addition, a recursive subroutine call is possible because a new storage area is assigned each time a subroutine is entered. This feature, together with in-line symbolic coding, provides powerful real time programming capability.

### Trace Statement

A TRACE STATEMENT is provided, permitting the tracing of certain variables or segments of a program. The TRACE is used at the source language level and can be used in two different ways. The first will generate a list of a specified set of variables and their values each time one of the variables is defined. The second will generate a list of the values of all variables appearing on the left side of an assignment statement, IF expressions evaluated, and the statement numbers of all statements executed in a particular segment of a program. Values listed include the name of the variable or a designation specifying the listed value as an IF expression result or statement number. The TRACE facility means the FORTRAN programmer can debug at the source language level without having to resort to octal memory dumps or console debugging.

### Memory Map

A memory map is provided at compile time on an optional basis. The memory map is a listing of all variable names, array names, and constants used by the program with their relative address assignments. The listing will include all subroutines called and last location where called.

### Error Diagnostics

Comprehensive error diagnostics are provided. The erroneous part of the statement is clearly marked. Whenever possible, the entire statement is processed when an error has been detected. The compiler will always continue to the end of the program. Thus, complete error diagnostics may be obtained in one compilation.

### Program Linking

If a program is too large to be stored in memory, the programmer can divide it into "links" by means of ADVANCE Series FORTRAN II LINK statement. At run time, routines in the MONITOR system automatically handle the execution of the segments of the linked program.

### Assembly Language Output

A symbolic assembly language listing of the binary object program output of the compiler is optional at compile time. The listing contains the symbolic instructions equivalent to the binary code output from the compiler. This assembly language output listing is useful as a debugging aid. By including certain pseudo operations codes in in-line assembly language, the assembly language output can be assembled by the assembler. This will allow modification of programs at the assembly language level.

### Compiler Input/Output

The source language input to the compiler can be on paper tape, cards, magnetic tape, or the on-line typewriter. The output of the compiler is a binary executable object program. The output of the compiler may be paper tape, cards, or magnetic tape with the option of obtaining a source language listing on the on-line printer or the on-line typewriter. The memory map is listed on the same device as the source language listing. The assembly language listing may be on any output device other than that used for the binary output. The compiler operates as part of the ADVANCE Series MONITOR system. All input/output device assignments can be easily changed at compile time.

### Operation

The compiler requires only one-pass over the FORTRAN II source program to produce a relocatable binary object program. The compiler will operate in a system consisting of an ASI ADVANCE Series Central Processor with 4,096 words of memory (minimum), a listing device, and either card or paper tape input/output. A FORTRAN main program can be compiled together with all the needed FORTRAN function of subroutine subprograms or they may be compiled separately. Binary relocatable subroutines assembled or compiled previously can be automatically loaded with the binary object program.

## ADVANCE SERIES EXTENDED FORTRAN II STATEMENT TYPES

DIMENSION  
GO TO n  
GO TO n, (n<sub>1</sub>, n<sub>2</sub>, n<sub>3</sub>, ...)  
GO TO (n<sub>1</sub>, n<sub>2</sub>, ...), i  
a = b  
ASSIGN i to n  
IF(a) n<sub>1</sub>, n<sub>2</sub>, n<sub>3</sub>  
IF (L) S  
SENSE LIGHT i  
IF (SENSE LIGHT i) n<sub>1</sub>, n<sub>2</sub>  
IF (SENSE SWITCH i) n<sub>1</sub>, n<sub>2</sub>  
IF ACCUMULATOR OVERFLOW n<sub>1</sub>, n<sub>2</sub>  
IF QUOTIENT OVERFLOW n<sub>1</sub>, n<sub>2</sub>  
IF DIVIDE CHECK n<sub>1</sub>, n<sub>2</sub>  
PAUSE n  
STOP n  
DO n i = m<sub>1</sub>, m<sub>2</sub>, m<sub>3</sub>  
CONTINUE

FORMAT  
EQUIVALENCE  
FREQUENCY  
LINK n, m  
COMMON  
RETURN  
FUNCTION name (arg<sub>1</sub>, arg<sub>2</sub>, ...)  
SUBROUTINE name (arg<sub>1</sub>, arg<sub>2</sub>, ...)  
CALL name (arg<sub>1</sub>, arg<sub>2</sub>, ...)  
DATA d<sub>1</sub>/v<sub>1</sub>/, d<sub>2</sub>/v<sub>2</sub>/, ...  
REAL  
INTERGER  
LOGICAL  
READ n, list  
ACCEPT n, list  
ACCEPT TAPE n, list  
READ INPUT TAPE i, n, list  
READ (u, n) list

PUNCH n, list  
PUNCH TAPE n, list  
TYPE n, list  
PRINT n, list  
WRITE (u, n) list  
WRITE TAPE i, list  
WRITE (u) list  
WRITE OUTPUT TAPE i, n, list  
READ TAPE i, list  
READ (u) list  
END FILE i  
REWIND i  
BACKSPACE i  
TRACE n  
TRACE list  
EOF



## ADVANCE Series Symbolic Translator—ASIST

ASI's Symbolic Translator, ASIST, is an assembly program that provides a powerful, flexible programming tool. ASIST features:

### ASSEMBLY CONTROL

- Select Input, Output, and Listing Media
- Specify Title and Date on Listing
- Symbol Table Control
- Symbol Table Listing
- Assembler-directed IF Statement
- Listing Control
- Link Specification
- Fixed or Relocatable Origin of Program

### PSEUDO OPERATIONS

- Symbolic Subroutine Entry Point Specifications
- EXTERNAL References
- Data Allocation (DEC, OCT, RES, ALF, CMN, etc.)
- Object Program Equipment Specifications

### MACRO STATEMENTS

- Programmed-defined Macros
- Input/Output Macros
- Floating Point Macros
- Arithmetic Routine Macros
- Subroutine CALL Macros

### OTHER

- One-Pass Operation with Optional Two-Pass
- Literal Usage
- CMN (COMMON) compatibility with FORTRAN II

Some of the highlights of the ADVANCE Series SYMBOLIC MACRO ASSEMBLER are:

**EQP Statement:** Input and Output to an assembled program may be specified and/or changed at run time by the EQP statement. This selects the logical equipment to be used and can reassign the equipment to any legal peripheral device.

**If Statement Assembly Control:** This feature allows for bypassing sections of an object program at assembly time under control of external indications such as the sense switches.

**Symbol Table Control:** Symbols which have been defined and used, but are no longer required . . . may be deleted from the symbol table. This allows room for new symbols. Thus, a very large program can be assembled with a relatively small symbol storage area.

**Listing Control:** The assembly language listing output from ASIST can be controlled by page EJECT and line SPACE commands. This yields a better documented listing.

**External References:** Entry points, constants, data areas, and commands in one program can be referenced by another independently assembled program. The EXT and ENTRY pseudo operations are used to accomplish this purpose.

**Programmer Defined Macros:** Segments of coding, which are used frequently throughout a program, can be defined at the beginning and referenced by a mnemonic code with parameters. This increases coding efficiency and readability of the program.

**CMN (Common) Compatibility with Fortran II:** The COMMON statement of ASIST allows easy specification of FORTRAN II Type Common Declarations. ASIST and FORTRAN common assignments are identical. This allows subroutines and links to be written in either language.

**One-Pass Operation:** Only One-Pass is made over the source language yielding a binary program. This is true regardless of the Input/Output configuration. For users without magnetic tape units, this is especially useful. They do not have to reload cards or paper tape for a second pass. However, a Two-Pass Option is included for more comprehensive listings and smaller binary decks.

**Subroutine Linkage:** Flexibility in subroutine handling is of prime importance for efficient operation of a software system. ASIST provides capabilities that allow complete flexibility in subroutine linkage. Pseudo operations such as ENTRY and ORG provide the information to the assembler necessary to generate relocatable binary subroutines that can be linked with other programs including FORTRAN II programs.

**Program Linking:** If a program is too large to be stored in memory, the programmer can divide it into "links" by means of the pseudo operation "LINK." This facility allows a program to be segmented. Special link handling routines are provided which automatically control the loading and execution of the various segments in a linked program.

**ASIST Operation:** The assembler will operate in a system consisting of an ASI ADVANCE Series Central Processor with 4,096 words of memory (minimum), a listing device, and either card or paper tape input/output. Some of the extensive options, however, restrict the size of the symbol table in the minimum system and these can be omitted at the discretion of the user. An ASIST main program can be assembled together with all needed subroutine subprograms or they may be assembled separately.



## ADVANCE Series Operating System—MONITOR

The ASI MONITOR System for ADVANCE Series facilitates job processing. It simplifies programming by offering the following:

### JOB PROCESSING

- Batch processing on card systems
- Automatic assignment of peripheral equipment
- Operator control via the online typewriter
- Initiates compilations and assemblies; loads; links; and initiates execution of subprograms
- Loads and initiates execution of utility programs
- Automatic "Compile and Execute" on one mag. tape system
- Automatic "Assemble and Execute" on one mag. tape system

### DEBUGGING AIDS

- Complete diagnostics on operator control information
- Special debugging dumps like PDUMP
- Recovery dumps

### INPUT/OUTPUT CONTROL AND SPECIAL REQUESTS

- Input/Output Routines
- External Interrupt Control
- Tape Handling
- Internal Interrupt Control
- Sampling of Equipment Status
- Date Specification

The programmer specifies monitor operations with control statements and with requests. These are punched in cards or entered via the on-line typewriter.

The MONITOR exercises primary control of the routines which compose the ASI ADVANCE Series operating system. It is this operating system which turns the ADVANCE Series into a flexible tool allowing the user to achieve maximum use of the hardware's advanced design features. Some of the important aspects of the ADVANCE Series operating system are:

**Program Linking:** Special operations are available to the programmer for specifying to the compiler or assembler that a program is to be segmented. The MONITOR system contains various routines to assist in the preparation and execution of these segmented programs. The segments of a program, which are contained on a magnetic tape, are automatically loaded and executed as they are called for by the program.

**Load and Go:** The MONITOR system contains various control routines which monitor the compilation, assembly, and execution of a program. With a one magnetic tape unit system, "Load and Go" operation is possible with FORTRAN and assembly language programs. The operator merely instructs the system that "Load and Go" is desired. The source card deck (or paper tape) is placed in the reader.

After depressing the run button, operation proceeds automatically until the program has completed its execution.

**Modularity:** The ASI ADVANCE Series MONITOR system is capable of utilizing various equipment configurations. The operator can specify the particular equipment that is available and the system will adjust itself to accommodate the specified configuration. The system can utilize line printers, magnetic tape, cards, paper tape and the on-line typewriter.

**Job Stacking:** The MONITOR system has the facilities for batch processing both FORTRAN compilations and assemblies.

Assemblies or compilations can be stacked behind one another. After the operator specifies the operation, assembling or compiling proceeds automatically through one job after another until an end-of-run card is encountered signifying the end of the run.

**Utility Routines:** A library of utility routines are available to the operator on the master tape. These are such items as memory dumps, conversion routines, debugging aids, card-to-tape programs, tape-to-printer programs, etc.

All utility routines are relocatable and can be loaded anywhere in memory. The operator, via the on-line typewriter, directs the system specifying which routine is needed and where it is to be loaded.

**Common Storage:** FORTRAN programs and ASIST programs may specify that data storage is to occupy a common area. This allows programs to share temporary storage, and FORTRAN coded programs to communicate with assembly coded routines by using a common data storage.

The system loader automatically allocates common area when it is loading programs. Common storage occupies the same memory space during execution as the system loader does when loading programs. This allows effective utilization of memory storage since the space taken up by the loader can be used by the programs during execution.

**Mathematical and Utility Routines:** ASI provides a complete set of single and double-precision arithmetic subroutines. This includes all the standard floating point operations in single and double precision as well as double precision integer arithmetic. Standard FORTRAN functions such as sine, cosine, etc., are available for both single and double precision floating point. In addition a comprehensive set of utility routines is available, allowing tape reproduction, conversion of cards to magnetic tape, magnetic tape to cards, line printer memory dump, etc.

ASI maintains a programming facility which is currently adding to and improving on our library of mathematical routines and utility programs. Thus ASI's customers are guaranteed an ever expanding program library, in accordance with ASI's policy of new program development.



## Support

An ADVANCE Series programming workshop is provided at regular intervals to provide additional assistance and training for special problems encountered by customer programmers.

Training courses for operators are conducted at both the ASI facility and at the customer's location during equipment installment.

When requested, ASI will provide a special familiarization course designed to inform company management of the particular procedures associated with the use of a computer system.

Trained field applications analysts are available to you on a local level to aid you in the running routine of the system and in developing new approaches to your programming problems. They will provide necessary assistance and suggest procedures in order to obtain maximum use and efficiency from your system.

The programming staff of ASI is constantly striving for better and faster methods of solving problems. When new solutions are developed, ASI will provide you with the new systems at no cost and provide training courses in their applications and implementation. A similar type of benefit is available from ASI's Users Group, an organization of users of ASI's equipment which provides an exchange of methods and programs used for all types of applications.

In addition to the programming systems described in this brochure, ASI offers a complete support program which includes pre-delivery assistance in programming

and debugging, training courses for all levels of programmers, operators, and company management, and post-delivery support by applications specialists who provide the necessary assistance to keep your system operating smoothly and efficiently.

ASI will provide the use of our facilities and the assistance of our programming staff before delivery of your system so that you may begin productive use of the system as soon as it is installed. This service is included as part of the system package at no additional cost. When feasible, at a contracted cost, ASI will write and debug special programs required to solve your problems.

Included in the training courses offered by ASI is the basic FORTRAN class which covers FORTRAN II. Topics discussed include: the expression of data, logical, and input/output statements as exemplified through the use of numerical methods. This course is primarily intended for the novice FORTRAN programmer.

An introductory course in ADVANCE Series machine language programming covers the ADVANCE Series computer specifications and a complete description of the central processor including information representation, instruction format; operating instructions and basic machine language operations. This course provides instruction in the preparation of ADVANCE Series symbolic programs including language format, declarations, and basic structure of symbolic program and subroutines.

### Fortran Programming Example

ASI FORTRAN CODING FORM	
PROGRAM COMPUTE ROOTS OF QUADRATIC EQUATION	
ROUTINE	
STATEMENT NO.	FORTRAN STATEMENT
1	C
2	1
3	2
4	3
5	4
6	5
7	6
8	7
9	8
10	9
11	10
12	11
13	12
14	13
15	14
16	15
17	16
18	17
19	18
20	19
21	20
22	21
23	22
24	23
25	24
26	25
27	26
28	27
29	28
30	29
31	30
32	31
33	32
34	33
35	34
36	35
37	36
38	37
39	38
40	39
41	40
42	41
43	42
44	43
45	44
46	45
47	46
48	47
49	48
50	49
51	50
52	51
53	52
54	53
55	54
56	55
57	56
58	57
59	58
60	59
61	60
62	61
63	62
64	63
65	64
66	65
67	66
68	67
69	68
70	69
71	70
72	71
73	72
74	73
75	74
76	75
77	76
78	77
79	78
80	79
81	80
82	81
83	82
84	83
85	84
86	85
87	86
88	87
89	88
90	89
91	90
92	91
93	92
94	93
95	94
96	95
97	96
98	97
99	98
100	99
101	100
102	101
103	102
104	103
105	104
106	105
107	106
108	107
109	108
110	109
111	110
112	111
113	112
114	113
115	114
116	115
117	116
118	117
119	118
120	119
121	120
122	121
123	122
124	123
125	124
126	125
127	126
128	127
129	128
130	129
131	130
132	131
133	132
134	133
135	134
136	135
137	136
138	137
139	138
140	139
141	140
142	141
143	142
144	143
145	144
146	145
147	146
148	147
149	148
150	149
151	150
152	151
153	152
154	153
155	154
156	155
157	156
158	157
159	158
160	159
161	160
162	161
163	162
164	163
165	164
166	165
167	166
168	167
169	168
170	169
171	170
172	171
173	172
174	173
175	174
176	175
177	176
178	177
179	178
180	179
181	180
182	181
183	182
184	183
185	184
186	185
187	186
188	187
189	188
190	189
191	190
192	191
193	192
194	193
195	194
196	195
197	196
198	197
199	198
200	199
201	200
202	201
203	202
204	203
205	204
206	205
207	206
208	207
209	208
210	209
211	210
212	211
213	212
214	213
215	214
216	215
217	216
218	217
219	218
220	219
221	220
222	221
223	222
224	223
225	224
226	225
227	226
228	227
229	228
230	229
231	230
232	231
233	232
234	233
235	234
236	235
237	236
238	237
239	238
240	239
241	240
242	241
243	242
244	243
245	244
246	245
247	246
248	247
249	248
250	249
251	250
252	251
253	252
254	253
255	254
256	255
257	256
258	257
259	258
260	259
261	260
262	261
263	262
264	263
265	264
266	265
267	266
268	267
269	268
270	269
271	270
272	271
273	272
274	273
275	274
276	275
277	276
278	277
279	278
280	279
281	280
282	281
283	282
284	283
285	284
286	285
287	286
288	287
289	288
290	289
291	290
292	291
293	292
294	293
295	294
296	295
297	296
298	297
299	298
300	299
301	300
302	301
303	302
304	303
305	304
306	305
307	306
308	307
309	308
310	309
311	310
312	311
313	312
314	313
315	314
316	315
317	316
318	317
319	318
320	319
321	320
322	321
323	322
324	323
325	324
326	325
327	326
328	327
329	328
330	329
331	330
332	331
333	332
334	333
335	334
336	335
337	336
338	337
339	338
340	339
341	340
342	341
343	342
344	343
345	344
346	345
347	346
348	347
349	348
350	349
351	350
352	351
353	352
354	353
355	354
356	355
357	356
358	357
359	358
360	359
361	360
362	361
363	362
364	363
365	364
366	365
367	366
368	367
369	368
370	369
371	370
372	371
373	372
374	373
375	374
376	375
377	376
378	377
379	378
380	379
381	380
382	381
383	382
384	383
385	384
386	385
387	386
388	387
389	388
390	389
391	390
392	391
393	392
394	393
395	394
396	395
397	396
398	397
399	398
400	399
401	400
402	401
403	402
404	403
405	404
406	405
407	406
408	407
409	408
410	409
411	410
412	411
413	412
414	413
415	414
416	415
417	416
418	417
419	418
420	419
421	420
422	421
423	422
424	423
425	424
426	425
427	426
428	427
429	428
430	429
431	430
432	431
433	432
434	433
435	434
436	435
437	436
438	437
439	438
440	439
441	440
442	441
443	442
444	443
445	444
446	445
447	446
448	447
449	448
450	449
451	450
452	451
453	452
454	453
455	454
456	455
457	456



# ASI

8001 Bloomington Freeway, Minneapolis, Minnesota 55420 / Division Electro-Mechanical Research, Inc.

**SALES OFFICES** BOSTON 272-5400, 1 Garfield Circle, Burlington • COCOA BEACH, FLA. 783-4905, Suite 1, Holiday Office Center, 1325 N. Atlantic Avenue • DALLAS DI 8-4170, Room 104, First Federal Building, 440 Northlake Center • DENVER SU 9-1834, 3600 S. Lincoln, Englewood • EL PASO 751-2344, 8888 Dyer St. • HOUSTON MI 4-1856, 7135 Office City Drive • HUNTSVILLE, ALA. 881-4822, 3322 S. Memorial Parkway • LOS ANGELES ST 2-7030, 15551 Cabrito Road, Van Nuys • MINNEAPOLIS 888-9581, 8001 Bloomington Freeway • NEW YORK MU 6-4762, 475 Fifth Avenue • SUNNYVALE, CALIF. 245-3694, 505 W. Olive • WASHINGTON, D. C. 864-6340, 5012 College Avenue, College Park, Md.